

# Towards Adaptive Hidden Markov Model Beamformers

John McDonough, Dominik Raub, Matthias Wölfel and Alex Waibel

**Abstract**—Recent interest within the automatic speech recognition community has focused on the recognition of “meeting room” speech. In this scenario, the microphone is located in the medium field, rather than being mounted on a headset and positioned next to the speaker’s mouth. Hence, this is a natural application for arrays of microphones and beamforming techniques. Recent research has indicated that a maximum likelihood (ML) criterion, evaluated with respect to a hidden Markov model, can be used to estimate optimal filter coefficients for a filter-and-sum beamformer. Indeed, this approach has shown significant reductions in word error rate with respect to a standard delay-and-sum beamformer. Because acoustic conditions—e.g., speaker locations, noise levels, reverberation characteristics—change constantly in a meeting room environment, a beamformer must continuously adapt to provide optimal performance. This work is an investigation into adaptive beamforming algorithms based on the ML criterion. In particular, we show how a recursive version of the well-known expectation maximization algorithm can be used to set the sensor weights of a subband domain generalized sidelobe cancelling beamformer. We derive two re-estimation algorithms: The first is a simple probabilistic gradient descent procedure. The second is an instance of an extended Kalman filter.

**Index Terms**—microphone arrays, beamforming, Kalman filters, speech recognition

## I. INTRODUCTION

INTEREST within the automatic speech recognition (ASR) research community has recently focused on the recognition of “meeting room” speech. In this scenario, the microphone is located in the medium field, rather than being mounted on a headset and positioned next to the speaker’s mouth. Hence, this is a natural application for arrays of microphones and beamforming techniques. For two principal reasons, however, conventional beamforming techniques are not well-suited to this application: Firstly, conventional array processing algorithms are based on the assumption that the desired signal (i.e., a user’s speech) is uncorrelated with any noise or interference that is also present in the environment. For ASR applications, this assumption is unjustified, as the most important source of signal distortion is room reverberation, which consists of multiple delayed versions of the original signal, and hence is highly correlated with it. Secondly, conventional array processing algorithms typically maximize a quadratic criterion such as signal-to-noise ratio (SNR), while simultaneously satisfying a constraint that any signal from a desired direction pass undistorted through the

system. Experience has shown, however, that quadratic criteria such as SNR are only weakly correlated with the recognition accuracy of an ASR system; improvements in SNR often do not translate into reductions in word error rate (WER).

Recent work by Seltzer [1] has provided further evidence that the conventional beamforming algorithms mentioned above are not optimal for ASR. In the ASR field, the most important maximization criterion for a wide variety of parameter estimation tasks is the maximum likelihood (ML) criterion. Modern speech recognizers are based on the hidden Markov model (HMM). Seltzer’s findings indicate that an ML criterion, evaluated with respect to an HMM such as is typically used for ASR, can be used to estimate optimal filter coefficients for a filter-and-sum beamformer [1]. Indeed, this approach has shown significant reductions in WER with respect to a conventional delay-and-sum beamformer.

Because acoustic conditions—e.g., speaker locations, noise levels, reverberation characteristics—change constantly in a meeting room environment, a beamformer must continuously adapt to provide optimal performance. In this work, we present a number of adaptive beamforming algorithms based on ML optimization criteria, where, following Seltzer [1], the likelihood is evaluated with respect to a HMM. In particular, we will show how a recursive version of the well-known expectation maximization (EM) algorithm can be used to set the sensor weights of a frequency domain generalized sidelobe cancelling beamformer. The approaches we propose as well as the organization of the balance of this work can be described as follows.

Typically, ML estimation for a HMM is done with the EM algorithm. The conventional EM algorithm is unsuitable for our application, however, because it updates the parameter values only after cycling through *all* available training data, which we refer to as the *unbounded delay* problem. Moreover, the EM algorithm requires that all training data from the distant past be retained, which we refer to as the *growing data* problem. In Section II-B, we show how the unbounded delay problem can be avoided through recourse to the *recursive* expectation-maximization (REM) algorithm. In the REM algorithm, the parameters of interest are updated as soon as a new sample arrives. Section II also develops the form of the EM auxiliary function to be used in the remaining sections.

In Section III-A, we review the definition of the *cepstral sequence*, which is used as an input feature in most modern ASR systems. In Section III-B we introduce the assumption that the cepstral features are derived from the subband domain output of a generalized sidelobe cancelling (GSC) beamformer. Section III-C then uses the preceding development to derive

All authors are with the Interactive Systems Laboratories at the Universität Karlsruhe in Karlsruhe, Germany. Email: jmc@ira.uka.de, Web site: <http://isl.ira.uka.de/~jmc> The authors wish to thank Oliver Schrempf and Fabian Jakobs for their help in recording the data and developing the software used for the experiments reported in this work.

both a ML beamformer based on conjugate gradient optimization as originally proposed by Seltzer [1], as well as a simple least mean square (LMS) error-style algorithm. The LMS algorithm so obtained is remarkably similar to the conventional LMS algorithm.

Taking the GSC beamformer and definition of cepstral sequences from Section III-A as a starting point, we show in Section IV-A how the maximum likelihood HMM beamformer can be posed as a *nonlinear* least squares estimation problem, where the nonlinearity is due to the  $\log | - |^2$  factor appearing in the definition of cepstral sequences. We then linearize the estimator about the current operating point, and derive recursive update formulae very similar to those used in conventional recursive least squares (RLS) estimation. This approach effectively solves the growing data problem.

In Section IV-B we discuss the steps necessary to cast the linearized RLS beamforming algorithm developed in Section IV-A as an iterated extended RLS beamformer. This approach offers significant advantages in terms of speed of convergence.

Although the HMM beamformer derived in this work operates in the subband domain, the sensor weights for each subband *cannot* be estimated separately, as in conventional beamformers. This is a product of the nonlinearity mentioned above. In Section IV-C, we discuss possibilities for treating the subbands independently, and thereby obtaining the numerous ensuing advantages. Section IV-D briefly discusses the advantages inherent in a square-root implementation [2, §11] of the iterated extended RLS estimator derived previously. In conventional beamforming, it is often useful to apply diagonal loading to the observation correlation matrix in order to limit the size of the active weight vector, as this improves the final sidelobe structure and reduces the sensitivity of the beamformer to steering errors [3, §6.6]. Section IV-D also describes how diagonal loading can be added to the iterated extended RLS beamformer considered here.

Section V presents the results of several large vocabulary conversational speech recognition experiments that were conducted to test the effectiveness of the algorithms proposed in this work.

In the final section, we summarize the results of this work, and thereafter discuss our conclusions and plans for future work.

## II. MAXIMUM LIKELIHOOD ESTIMATION

As mentioned in the introduction, the preferred criterion for parameter estimation in most ASR applications is the ML criterion. In this section, we provide a brief introduction to the EM algorithm, which is invariably used whenever performing ML estimation in conjunction with a HMM. Thereafter we introduce the REM algorithm, which is perhaps not so well-known but will prove vital for the development in subsequent sections. We will then briefly discuss the specialization of both algorithms for use with a HMM.

### A. The Expectation Maximization (EM) Algorithm

To begin, let us define an *observation*  $\mathbf{y}$  drawn from a set of training data  $\mathcal{Y}$ . Although  $\mathbf{y}$  is what we actually observe,

it is useful to think of  $\mathbf{y}$  as being “incomplete,” inasmuch as it does not contain certain useful information. Let  $\mathbf{x}$  be the *complete observation* associated with  $\mathbf{y}$ . In addition to all the information in  $\mathbf{y}$ ,  $\mathbf{x}$  also contains this missing information, the so-called *hidden variables*.

Suppose that  $f(\mathbf{y}; \Lambda)$  is a probability density function (pdf) on  $\mathbf{y}$  specified by some set of parameters  $\Lambda$ . We desire to obtain a ML estimate of  $\Lambda$  given the training set  $\mathcal{Y} = \{\mathbf{y}_t\}$ , such that

$$\hat{\Lambda} = \arg \max_{\Lambda} \log f(\mathcal{Y}; \Lambda)$$

Assuming that all observations  $\mathbf{y}_i$  are independent and identically distributed (iid), we can equivalently write

$$\hat{\Lambda} = \arg \max_{\Lambda} \sum_t \log f(\mathbf{y}_t; \Lambda)$$

The EM algorithm as originally proposed by Dempster, Laird, and Rubin [4] proceeds in two steps. In the E- or *expectation*-step, we evaluate the *auxiliary function*  $Q(\Lambda, \Lambda^{(i)})$ , given by

$$Q(\Lambda, \Lambda^{(i)}) = \sum_t \mathcal{E} \left\{ \log f(\mathbf{x}_t, \Lambda) | \mathbf{y}_t; \Lambda^{(i)} \right\}$$

where  $\Lambda^{(i)}$  is the current parameter estimate; i.e., the estimate after  $i$  iterations. Note that the expectation above is over the complete observation  $\mathbf{x}_t$ , and is evaluated with pdf parameters  $\Lambda^{(i)}$ . In the M- or *maximization*-step, we update the parameter estimate according to

$$\Lambda^{(i+1)} = \arg \max_{\Lambda} Q(\Lambda, \Lambda^{(i)})$$

Provided the parameter estimates have not yet converged, the algorithm then continues with another E-step.

The proof that the EM algorithm converges is based on the inequality

$$Q(\Lambda, \Lambda^{(i)}) - Q(\Lambda^{(i)}, \Lambda^{(i)}) < \log f(\mathcal{Y}; \Lambda) - \log f(\mathcal{Y}; \Lambda^{(i)}) \quad (\text{II.1})$$

from which it follows [5, §9.2]

$$Q(\Lambda, \Lambda^{(i)}) > Q(\Lambda^{(i)}, \Lambda^{(i)}) \Rightarrow f(\mathcal{Y}; \Lambda) > f(\mathcal{Y}; \Lambda^{(i)})$$

### B. The Recursive Expectation Maximization (REM) Algorithm

We now present a recursive version of the EM algorithm originally proposed by Titterton [6], and further refined by Frenkel and Feder [7]. As before, the algorithm consists of two steps. In the E-step, the auxiliary function

$$\begin{aligned} Q(\Lambda | \Lambda^{(t)}) &= L^{(t+1)}(\Lambda) \\ &= \sum_{i=1}^{t+1} \lambda^{t+1-i} \cdot \mathcal{E} \left\{ \log f(\mathbf{x}_i; \Lambda) | \mathbf{y}_i; \Lambda^{(i-1)} \right\} \quad (\text{II.2}) \\ &= \lambda L^{(t)}(\Lambda) + \mathcal{E} \left\{ \log f(\mathbf{x}_{t+1}; \Lambda) | \mathbf{y}_{t+1}; \Lambda^{(t)} \right\} \quad (\text{II.3}) \end{aligned}$$

is evaluated, where  $\lambda \in (0, 1]$  is the so-called *forgetting factor*. The statistical expectation over each complete observation  $\mathbf{x}_{t+1}$  is evaluated *once* using the parameter values  $\Lambda^{(t)}$  that are current when  $\mathbf{x}_{t+1}$  arrives. The M-step is then the same as for the regular EM algorithm,

$$\Lambda^{(t+1)} = \arg \max_{\Lambda} Q(\Lambda, \Lambda^{(t)}) \quad (\text{II.4})$$

Note that, in keeping with the recursive nature of the algorithm, each parameter estimate in (II.4) is associated with a time  $t$ .

### C. EM Algorithm for the Hidden Markov Model

Consider a directed graph representing an HMM comprised of a set  $\{n_i\}$  of *nodes* and a set  $\{e_{j|i}\}$  of *edges*, where  $n_i$  is the  $i$ -th node and  $e_{j|i}$  is the directed edge from  $n_i$  to  $n_j$ . Let us define the *transition probability*

$$p_{j|i} = P(n_j|n_i) = P(e_{j|i}|n_i) \quad (\text{II.5})$$

Let  $g_{ik}$  denote the  $k$ -th Gaussian component associated with node  $n_i$  and define the *mixture weight*

$$q_{k|i} = P(g_{ik}|n_i)$$

The conditional likelihood assigned to a single observation  $\mathbf{y}$  by the Gaussian mixture model associated with node  $n_i$  can be expressed as

$$P(\mathbf{y}; \Lambda_i) = \sum_k q_{k|i} P(\mathbf{y}; \Lambda_{ik})$$

where  $\Lambda_i = \{(q_{k|i}, \Lambda_{ik})\}$ . Here  $\Lambda_{ik} = (\boldsymbol{\mu}_{ik}, \mathbf{Q}_{ik})$  and  $\boldsymbol{\mu}_{ik}$  and  $\mathbf{Q}_{ik}$  respectively denote the mean vector and covariance matrix. The likelihood returned by the  $k$ -th Gaussian component is

$$P(\mathbf{y}; \Lambda_{ik}) = \frac{1}{\sqrt{|2\pi\mathbf{Q}_{ik}|}} \exp \left[ -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_{ik})^T \mathbf{Q}_{ik}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{ik}) \right]$$

Finally, let  $\mathbf{y}_1^T$  denote the *sequence* of observations associated with a given utterance and let  $\mathbf{y}_1^t$  for some  $1 \leq t \leq T$  denote a subsequence of  $\mathbf{y}_1^T$ .

Let us define the *forward probability* [8, §12.2]  $\alpha(\mathbf{y}_1^t, i)$  as the likelihood of generating the observation subsequence  $\mathbf{y}_1^t$  and arriving at state  $n_i$  at time  $t$  given the current model parameters  $\Lambda$ :

$$\alpha(\mathbf{y}_1^t, i) = P(n_t = n_i, \mathbf{y}_1^t; \Lambda) \quad (\text{II.6})$$

where—in a slight abuse of notation—we have used  $n_t$  to denote the HMM state associated with observation  $\mathbf{y}_t$ . Similarly, the *backward probability* is the likelihood of generating the observation subsequence  $\mathbf{y}_{t+1}^T$  conditioned on having started from state  $n_j$  at time  $t$ :

$$\beta(\mathbf{y}_{t+1}^T | j) = P(\mathbf{y}_{t+1}^T | n_t = n_j; \Lambda) \quad (\text{II.7})$$

These probabilities can be calculated via the well-known recursions [8, §12]:

$$\alpha(\mathbf{y}_1^{t+1}, j) = P(\mathbf{y}_{t+1}; \Lambda_j) \sum_i \alpha(\mathbf{y}_1^t, i) p_{j|i} \quad (\text{II.8})$$

$$\beta(\mathbf{y}_{t+1}^T | j) = \sum_i \beta(\mathbf{y}_{t+2}^T | i) p_{i|j} P(\mathbf{y}_{t+1}; \Lambda_i) \quad (\text{II.9})$$

Our interest in the forward-backward probabilities is due to the fact that they can be used to calculate the posterior probabilities

$$c_{jk,t} = P(g_t = g_{jk} | \mathbf{y}_1^T; \Lambda) \quad (\text{II.10})$$

$$= \frac{\alpha(\mathbf{y}_1^t, j) \beta(\mathbf{y}_{t+1}^T | j)}{P(\mathbf{y}_1^T; \Lambda)} \frac{q_{k|j} P(\mathbf{y}_t; \Lambda_{jk})}{\sum_m q_{m|j} P(\mathbf{y}_t; \Lambda_{jm})} \quad (\text{II.11})$$

During conventional HMM training, these posterior probabilities are necessary for mean and variance re-estimation, which is achieved by maximizing the auxiliary function

$$K(\Lambda | \Lambda^0) = -\frac{1}{2} \sum_{j,k,t} c_{jk,t} \left[ \log |2\pi\mathbf{Q}_{jk}| + (\mathbf{y}_t - \boldsymbol{\mu}_{jk})^T \mathbf{Q}_{jk}^{-1} (\mathbf{y}_t - \boldsymbol{\mu}_{jk}) \right] \quad (\text{II.12})$$

As we will find in Section III,  $K(\Lambda | \Lambda^0)$  can also be optimized with respect to the frequency-dependent sensor weights to perform ML beamforming.

Applying the REM algorithm to the HMM is somewhat more involved, inasmuch as each  $c_{jk,t}$  depends on all observations  $\mathbf{y}_1^T$  in a given utterance. Although it would be straightforward to update all parameters at the end of a complete utterance, updating more often than that, such as at the end of every word, would require an approximation of (II.8–II.9). For example, we might make a *Viterbi approximation* [8, §12.2] at the end states of words, and sum only along the most likely path.

## III. MAXIMUM LIKELIHOOD BEAMFORMING

Having completed our brief discussion of the EM and REM algorithms, we now take up the task of using this algorithm to perform ML beamforming. In so doing, we will introduce the concepts of cepstral sequences and the generalized sidelobe cancelling beamformer.

### A. Cepstral Sequences

Let  $\mathbf{v} = \{v_n\}$  denote a vector of cepstral coefficients associated with a vector  $\mathbf{V} = \{V_m\}$  of frequency samples, so that

$$v_n = \frac{1}{2M} \sum_{m=0}^{M-1} \log |V_m|^2 \cos \omega_m n \quad (\text{III.1})$$

where  $\omega_m = m/2\pi M$ . Defining the *cosine transform matrix*  $\mathbf{S} = \{S_{nm}\}$  whose components are given by

$$S_{nm} = \frac{1}{2M} \cos \frac{nm}{2\pi M}$$

for all  $n, m = 0, 1, \dots, M-1$ , we can rewrite (III.1) as

$$v_n = \sum_{m=0}^{M-1} S_{nm} \log |V_m|^2$$

If the features are to be used for ASR, a nonlinear *Mel-warpage* is typically applied to the frequency axis prior to the calculation of cepstral coefficients. In this case, we must replace the last equation with

$$v_n \triangleq \sum_{m=0}^{M-1} S_{nm} \log |\tilde{V}_m|^2 \quad (\text{III.2})$$

where

$$|\tilde{V}_m|^2 \triangleq \sum_l M_{ml} |V_l|^2 \quad (\text{III.3})$$

are the Mel-warped frequency or subband components and  $\mathbf{M} = \{M_{ml}\}$  is the Mel-warpage matrix.

### B. Generalized Sidelobe Cancelling Beamformers

Now consider the *generalized sidelobe cancelling* (GSC) beamformer [3, §6.7.3], whose final output is obtained from the difference of the outputs of upper and lower branches. On the upper branch, the input  $\mathbf{U}_m$  is multiplied by the *quiescent weight vector*  $\mathbf{w}_{q,m}$ , which is chosen to ensure the beamformer satisfies a *distortionless constraint*. The latter can be expressed as

$$\mathbf{w}_{q,m}^H \mathbf{g}_m = 1 \quad (\text{III.4})$$

where  $\mathbf{g}_m$  is the *array manifold vector* [3, §2.2], given by

$$\mathbf{g}_m = [e^{-j\omega_m \tau_0} \quad e^{-j\omega_m \tau_1} \quad \dots \quad e^{-j\omega_m \tau_{N-1}}]^T$$

In the above,  $N$  is the number of sensors in the array,  $\tau_n$  is the propagation delay between the source and the  $n$ -th sensor, and  $\omega_m$  is the center frequency of the  $m$ -th subband. On the lower branch, the input is first multiplied by the *blocking matrix*  $\mathbf{B}_m$ , which must be orthogonal to  $\mathbf{w}_{q,m}$ , such that

$$\mathbf{w}_{q,m}^H \mathbf{B}_m = 0$$

Thereafter, the output of the blocking matrix,

$$\mathbf{Z}_m = \mathbf{B}_m \mathbf{U}_m \quad (\text{III.5})$$

is multiplied by the *active weight vector*  $\mathbf{w}_{a,m}$ . The orthogonality of  $\mathbf{w}_{q,m}$  and  $\mathbf{B}_m$  ensures that the total weight vector  $\mathbf{w}_m = \mathbf{w}_{q,m} + \mathbf{B}_m \mathbf{w}_{a,m}$  satisfies  $\mathbf{w}_m^H \mathbf{g}_m = 1$  regardless of the value assigned to  $\mathbf{w}_{a,m}$ . In what follows, we will choose  $\hat{\mathbf{w}}_{a,m}$  according to the ML criterion, where the likelihood is calculated with respect to an HMM.

Assuming  $\mathbf{V}$  is the output of the GSC, we may write

$$V_m = (\mathbf{w}_{q,m}^H - \mathbf{w}_{a,m}^H \mathbf{B}_m^H) \mathbf{U}_m \quad (\text{III.6})$$

Setting  $\mathbf{w}_{a,m} = \mathbf{0}$  for all  $m$  we obtain the *delay and sum* beamformer, which is the baseline against which any adaptive beamforming algorithm must be judged. Equation (III.6) implies

$$|V_m|^2 = (\mathbf{w}_{q,m}^H - \mathbf{w}_{a,m}^H \mathbf{B}_m^H) \mathbf{U}_m \mathbf{U}_m^H (\mathbf{w}_{q,m} - \mathbf{B}_m \mathbf{w}_{a,m}) \quad (\text{III.7})$$

Equations (III.2) and (III.7) can be used to develop an expression for the gradient of  $v_n$  with respect to each of the  $\mathbf{w}_{a,m}$ . Taking a partial derivative on both sides of (III.2) gives

$$\frac{\partial v_n}{\partial \mathbf{w}_{a,m}^*} = \sum_{l=0}^{M-1} \frac{S_{nl} M_{lm}}{|\tilde{V}_l|^2} \cdot \frac{\partial |V_m|^2}{\partial \mathbf{w}_{a,m}^*} \quad (\text{III.8})$$

where  $\{|\tilde{V}_l|^2\}_m$  are the Mel-warped subband components (III.3). If we define

$$\rho_{nm}(t) \triangleq \sum_{l=0}^{M-1} \frac{S_{nl} M_{lm}}{|\tilde{V}_l|^2} \quad (\text{III.9})$$

then (III.8) can be rewritten as

$$\frac{\partial v_n}{\partial \mathbf{w}_{a,m}^*} = \rho_{nm}(t) \frac{\partial |V_m|^2}{\partial \mathbf{w}_{a,m}^*} \quad (\text{III.10})$$

We can take the desired derivative on both sides of (III.7) to obtain

$$\frac{\partial |V_m|^2}{\partial \mathbf{w}_{a,m}^*} = \mathbf{B}_m^H \mathbf{U}_m \mathbf{U}_m^H (\mathbf{B}_m \mathbf{w}_{a,m} - \mathbf{w}_{q,m}) \quad (\text{III.11})$$

Substituting (III.11) into (III.10), we arrive at

$$\frac{\partial v_n}{\partial \mathbf{w}_{a,m}^*} = \rho_{nm}(t) \cdot \mathbf{B}_m^H \mathbf{U}_m \mathbf{U}_m^H (\mathbf{B}_m \mathbf{w}_{a,m} - \mathbf{w}_{q,m}) \quad (\text{III.12})$$

### C. Two Simple Beamforming Algorithms

Our intention is to choose the active sensor weights  $\{\mathbf{w}_{a,m}\}$  so as to maximize the likelihood of the training set  $\{\mathbf{v}(t)\}_t$ . Hence, we set  $\Lambda = \{\mathbf{w}_{a,m}\}$ , and, in light of (II.12), seek to minimize the objective function

$$K(\Lambda | \Lambda^{(i-1)}) = \frac{1}{2} \sum_t [\mathbf{v}(t) - \boldsymbol{\mu}(t)]^T \mathbf{Q}^{-1}(t) [\mathbf{v}(t) - \boldsymbol{\mu}(t)] \quad (\text{III.13})$$

where  $\Lambda^{(i-1)}$  is the parameter estimate from the *previous* iteration. To reduce computation, we have assumed  $c_{jk,t} \equiv 1$  and let  $\boldsymbol{\mu}(t) = \boldsymbol{\mu}_{jk}$  and  $\mathbf{Q}(t) = \mathbf{Q}_{jk}$  denote the mean and variance associated with  $\mathbf{v}(t) = \mathbf{v}_t$ . This corresponds to the case wherein the sum is made only over the Viterbi path [8, §12.2], and there is a single Gaussian per HMM state. Based on (III.12–III.13) it is straightforward to develop a ML beamforming algorithm that uses a single, *global* utterance for enrollment, as proposed by Seltzer [1]. It is also straightforward to develop a simple LMS-style adaptive beamforming algorithm, wherein an instantaneous estimate of the gradient of the error surface is made, and then a small step is taken in the putative downhill direction.

Let  $K^{(t)}(\Lambda | \Lambda^{(i-1)})$  denote the contribution of the subband snapshot at time  $t$  to (III.13), so that

$$K^{(t)}(\Lambda | \Lambda^{(i-1)}) = \frac{1}{2} [\mathbf{v}(t) - \boldsymbol{\mu}(t)]^T \mathbf{Q}^{-1}(t) [\mathbf{v}(t) - \boldsymbol{\mu}(t)]$$

The partial derivative of  $K^{(t)}(\Lambda | \Lambda^{(i-1)})$  with respect to  $\mathbf{w}_{a,m}^*$  can be readily evaluated via the chain rule as

$$\frac{\partial K^{(t)}(\Lambda | \Lambda^{(i-1)})}{\partial \mathbf{w}_{a,m}^*} = -\nu_m(t) \cdot \mathbf{Z}_m(t) e_m^*(t) \quad (\text{III.14})$$

where  $\mathbf{Z}_m(t)$  is the output of the blocking matrix (III.5) and

$$e_m(t) \triangleq [\mathbf{w}_{q,m}^H - \hat{\mathbf{w}}_{a,m}^H(i-1) \mathbf{B}_m^H] \mathbf{U}_m(t) \quad (\text{III.15})$$

is the output of the beamformer for the snapshot  $\mathbf{U}_m(t)$  using the old sensor weights  $\hat{\mathbf{w}}_{a,m}(i-1)$ . In writing (III.14) we have also defined

$$\nu_m(t) \triangleq \sum_{n=0}^{L-1} \frac{v_n(t) - \mu_n(t)}{\phi_n(t)} \cdot \rho_{nm}(t) \quad (\text{III.16})$$

Using (III.14), we can readily calculate the gradient for an entire enrollment utterance as

$$K(\Lambda | \Lambda^{(i-1)}) = \sum_t \frac{\partial K^{(t)}(\Lambda | \Lambda^{(i-1)})}{\partial \mathbf{w}_{a,m}^*}$$

This gradient together with the actual value of  $K(\Lambda | \Lambda^{(i-1)})$  can be used to implement an optimization algorithm based on the method of conjugate gradients [9, §10.6]. Such a “global”

optimization procedure was proposed by Seltzer [1] for the beamforming application considered here.

Because a single utterance can last as long as a few dozen seconds, during which time the speaker may move, turn on a laptop, open a window, etc., it would be better to update the sensor weights after every *frame* instead of waiting until the end of an utterance. An adaptive beamformer which does exactly this can be obtained by considering the LMS update rule:

$$\hat{\mathbf{w}}_{a,m}(t) = \hat{\mathbf{w}}_{a,m}(t-1) - \xi_m(t) \cdot \frac{\partial K(t)(\Lambda|\Lambda^{(t-1)})}{\partial \mathbf{w}_{a,m}^*}$$

which, upon substituting (III.14), can be rewritten as

$$\hat{\mathbf{w}}_{a,m}(t) = \hat{\mathbf{w}}_{a,m}(t-1) + \xi_m(t) \cdot \nu_m(t) \mathbf{Z}_m(t) e_m^*(t) \quad (\text{III.17})$$

It is remarkable that (III.15–III.17) differ from the conventional LMS update rule for a GSC beamformer [3, §7.7.1.4] only by the factor  $\nu_m(t)$ . To set the step size  $\xi_m(t)$ , Van Trees [3, §7.7.2.2] recommends the heuristic

$$\xi_m(t) = \frac{\gamma}{\sigma_m^2(t)}$$

where

$$\sigma_m^2(t) = \beta \sigma_m^2(t-1) + (1-\beta) \mathbf{U}_m^H(t) \mathbf{U}_m(t)$$

is the average power in the  $m$ -th subband. In the above,  $\gamma$  is a constant with a typical value of  $0.005 < \gamma < 0.05$ . Similarly,  $\beta$  is a constant close to unity; i.e.,  $\beta \geq 0.99$ . We can also use a simple technique based on that in Van Trees [3, §7.7.4] for enforcing the quadratic constraint

$$\|\hat{\mathbf{w}}_{a,m}(t)\|^2 \leq \alpha^2$$

for some real  $\alpha > 0$ , which is beneficial in the presence of steering errors and other forms of mismatch [3, §6.6]. The algorithm first calculates

$$\tilde{\mathbf{w}}_{a,m}(t) = \hat{\mathbf{w}}_{a,m}(t-1) + \xi_m(t) \cdot \nu_m(t) \mathbf{Z}_m(t) e_m^*(t)$$

exactly as in (III.17). Thereafter, the final weight vector is obtained from

$$\hat{\mathbf{w}}_{a,m}(t) = \begin{cases} \tilde{\mathbf{w}}_{a,m}(t), & \text{if } \|\tilde{\mathbf{w}}_{a,m}(t)\|^2 \leq \alpha^2 \\ c_m(t) \tilde{\mathbf{w}}_{a,m}(t) & \text{otherwise} \end{cases} \quad (\text{III.18})$$

where, for a conventional LMS beamformer,

$$c_m(t) = \frac{\alpha}{\|\tilde{\mathbf{w}}_{a,m}(t)\|} \quad (\text{III.19})$$

In HMM beamforming, the sensor weights in each frequency bin *cannot* be optimized independently; hence, we found it beneficial to define a single scale factor  $c_m(t) = c(t)$  for all  $m$  according to

$$c(t) = \frac{\alpha}{\max_m \|\tilde{\mathbf{w}}_{a,m}(t)\|} \quad (\text{III.20})$$

#### IV. RECURSIVE LEAST SQUARES ESTIMATION

Here we provide the development necessary to cast the HMM beamformer as a recursive least squares estimation procedure.

##### A. Linearized Recursive Least Squares Estimation

As mentioned previously, our beamforming application involves a nonlinearity due to the  $\log | - |^2$  in the definition of cepstral sequences. The approach taken in this Section is to simply linearize this term about the current estimate of the sensor weights. Applying the matrix inversion lemma, we can then formulate a *linearized* recursive least squares (RLS) estimation algorithm.

Let  $N$  denote the number of sensors in the array, let  $M$  denote the number of frequency bands in the subband filter bank, and let  $L$  denote the length of the final cepstral sequence. Given the length  $N-1$  vector  $\mathbf{w}_{a,m}$  of active sensor weights corresponding to the  $m$ -th subband defined previously, let us now define the *stacked* active weight vector as

$$\mathbf{W}_a \triangleq [\mathbf{w}_{a,0} \quad \mathbf{w}_{a,1} \quad \cdots \quad \mathbf{w}_{a,M-1}]^T$$

As before, let  $\mathbf{U}_m(t)$  denote the array input at time  $t$  for the  $m$ -th subband, and define the matrix  $\mathbf{U}(t)$  of stacked inputs as

$$\mathbf{U}(t) \triangleq \text{diag} [\mathbf{U}_0(t) \quad \mathbf{U}_1(t) \quad \mathbf{U}_2(t) \quad \cdots \quad \mathbf{U}_{M-1}(t)]$$

Also define the stacked blocking matrix  $\mathbf{B}$  as

$$\mathbf{B} \triangleq \text{diag} [\mathbf{B}_0 \quad \mathbf{B}_1 \quad \mathbf{B}_2 \quad \cdots \quad \mathbf{B}_{M-1}]$$

As in Section III-C, we seek to minimize the auxiliary function

$$K(\Lambda|\Lambda^0) = \frac{1}{2} \sum_t [\mathbf{v}(t, \mathbf{W}_a) - \boldsymbol{\mu}(t)]^T \mathbf{Q}^{-1}(t) [\mathbf{v}(t, \mathbf{W}_a) - \boldsymbol{\mu}(t)] \quad (\text{IV.1})$$

where we have explicitly shown the dependence of  $\mathbf{v}(t) = \mathbf{v}(t, \mathbf{W}_a)$  on  $\mathbf{W}_a$ . It is then apparent that (IV.1) has the form of a *nonlinear* least squares estimation problem. As in (II.2–II.3), define the *exponentially-weighted squared error* as

$$\epsilon(t; \mathbf{W}_a) = \sum_{i=1}^t \lambda^{t-i} [\mathbf{v}(i, \mathbf{W}_a) - \boldsymbol{\mu}(i)]^H \mathbf{Q}^{-1}(i) [\mathbf{v}(i, \mathbf{W}_a) - \boldsymbol{\mu}(i)] \quad (\text{IV.2})$$

where  $\lambda$  is once more the forgetting factor. Next we seek to linearize this RLS estimator about the current estimate  $\hat{\mathbf{W}}_a(t)$  of the weight vector. From (III.12) and (III.15) it follows [10, §4]

$$\left[ \frac{\partial v_n}{\partial \mathbf{w}_{a,m}^*} \right]^H = C_{\text{RLS},nm}(t) \mathbf{Z}_m^H(t)$$

where

$$C_{\text{RLS},nm}(t) = -\rho_{nm}(t) \cdot e_m(t) \quad (\text{IV.3})$$

It is then apparent that  $\mathbf{v}(t; \mathbf{W}_a)$  can be approximated with the first order Taylor series,

$$\mathbf{v}(t; \mathbf{W}_a) \approx \mathbf{v}(t; \hat{\mathbf{W}}_a(t-1)) + C_{\text{RLS}}(t) \mathbf{Z}^H(t) [\mathbf{W}_a - \hat{\mathbf{W}}_a(t-1)] \quad (\text{IV.4})$$

where

$$C_{\text{RLS}}(t) = \{C_{\text{RLS},nm}(t)\}_{nm} \quad (\text{IV.5})$$

and

$$\mathbf{Z}(t) = \text{diag} [\mathbf{Z}_0(t) \quad \mathbf{Z}_1(t) \quad \mathbf{Z}_2(t) \quad \cdots \quad \mathbf{Z}_{M-1}(t)]$$

is the matrix of stacked blocking matrix outputs. We now apply this approximation to (IV.2), to obtain

$$\epsilon(t; W_a) = \sum_{i=1}^t \lambda^{t-i} \left[ C_{\text{RLS}}(t) Z^H(t) W_a - \bar{\mu}(i) \right]^H Q^{-1}(i) \cdot \left[ C_{\text{RLS}}(t) Z^H(t) W_a - \bar{\mu}(i) \right] \quad (\text{IV.6})$$

where

$$\bar{\mu}(t) = \mu(t) - \left[ v(t; \hat{W}_a(t-1)) - C_{\text{RLS}}(t) Z^H(t) \hat{W}_a(t-1) \right] \quad (\text{IV.7})$$

Through straightforward algebraic manipulations, (IV.6) can be expressed as

$$\epsilon(t; W_a) = \epsilon_0(t) - W_a^H \zeta(t) - \zeta^H(t) W_a + W_a^H \Phi(t) W_a \quad (\text{IV.8})$$

where

$$\begin{aligned} \Phi(t) &= \sum_{i=1}^t \lambda^{t-i} Z(i) C_{\text{RLS}}^H(i) Q^{-1}(i) C_{\text{RLS}}(i) Z^H(i) \\ &= \lambda \Phi(t-1) + Z(t) C_{\text{RLS}}^H(t) Q^{-1}(t) C_{\text{RLS}}(t) Z^H(t) \end{aligned} \quad (\text{IV.9})$$

$$\begin{aligned} \zeta(t) &= \sum_{i=1}^t \lambda^{t-i} Z(i) C_{\text{RLS}}^H(i) Q^{-1}(i) \bar{\mu}(i) \\ &= \lambda \zeta(t-1) + Z(t) C_{\text{RLS}}^H(t) Q^{-1}(t) \bar{\mu}(t) \end{aligned} \quad (\text{IV.10})$$

$$\begin{aligned} \epsilon_0(t) &= \sum_{i=1}^t \lambda^{t-i} \bar{\mu}^T(i) Q^{-1}(i) \bar{\mu}(i) \\ &= \lambda \epsilon_0(t-1) + \bar{\mu}^T(t) Q^{-1}(t) \bar{\mu}(t) \end{aligned}$$

Differentiating (IV.8) and equating the result to zero, we obtain the *normal equations*,

$$\Phi(t) W_a = \zeta(t)$$

which, assuming  $\Phi(t)$  is invertible, are readily solved as

$$\hat{W}_a(t) = \Phi^{-1}(t) \zeta(t) \quad (\text{IV.11})$$

Applying the *matrix inversion lemma* [2, §9.2] to (IV.9) provides

$$\begin{aligned} \Phi^{-1}(t) &= \lambda^{-1} \Phi^{-1}(t-1) - \lambda^{-2} \Phi^{-1}(t-1) Z(t) C_{\text{RLS}}^H(t) \\ &\quad \cdot \left[ Q(t) + \lambda^{-1} C_{\text{RLS}}(t) Z^H(t) \Phi^{-1}(t-1) Z(t) C_{\text{RLS}}^H(t) \right]^{-1} \\ &\quad \cdot C_{\text{RLS}}(t) Z^H(t) \Phi^{-1}(t-1) \end{aligned} \quad (\text{IV.12})$$

Defining the *precision matrix*  $P(t) = \Phi^{-1}(t)$  we can rewrite (IV.12) as

$$P(t) = \lambda^{-1} P(t-1) - \lambda^{-1} G_{\text{RLS}}(t) C_{\text{RLS}}(t) Z^H(t) P(t-1) \quad (\text{IV.13})$$

where

$$\begin{aligned} G_{\text{RLS}}(t) &= \lambda^{-1} P(t-1) Z(t) C_{\text{RLS}}^H(t) \\ &\quad \cdot \left[ Q(t) + \lambda^{-1} C_{\text{RLS}}(t) Z^H(t) P(t-1) Z(t) C_{\text{RLS}}^H(t) \right]^{-1} \end{aligned} \quad (\text{IV.14})$$

is the *Kalman gain* for this RLS problem. It is then straightforward to show that the desired update formulae for the active sensor weights is

$$\hat{W}_a(t) = \hat{W}_a(t-1) + G_{\text{RLS}}(t) \xi(t) \quad (\text{IV.15})$$

where

$$\xi(t) = \mu(t) - v(t; \hat{W}_a(t-1)) \quad (\text{IV.16})$$

is the *a priori estimation error*.

## B. Extended Recursive Least Squares Estimation

In RLS estimation, we calculate an estimate of a set of *deterministic* parameters so as to minimize a squared error criterion. In Kalman filtering, on the other hand, we estimate the current state of a stochastic process, which is itself assumed to be a *random* vector. Although the two estimators are formulated differently, the quantities estimated by each can be put in a one-to-one correspondence, as originally proposed in [11], and discussed in [2, §10.8]. This correspondence follows upon associating the parameters of interest in the RLS estimation problem with the state of the Kalman filter, whereupon the precision matrix  $P(t)$  for the RLS estimator can be equated to the covariance matrix of the *state estimation error* for the Kalman filter. Indeed, this correspondence is extremely useful, as it implies that by formulating a given RLS estimation problem as a problem in Kalman filtering, we can draw upon the vast literature on Kalman filtering published since Kalman's original paper [12] to improve the numerical robustness, rate of convergence, tracking capabilities and other characteristics of the estimator so obtained. In [10], this correspondence is developed in great detail for the HMM beamforming problem treated here. For reasons of brevity, we can only summarize the key points presented in that earlier work. The desired estimator can be derived with the following steps:

1. Cast the linearized RLS estimator developed in Section IV-A as an extended Kalman filter (EKF), wherein the non-linear observation equation is linearized about each new estimate of the sensor weights. This development is very similar to that in [2, §10.8].
2. Refine the EKF as an *iterated extended* Kalman filter (IEKF) [13, §8.3]. In the IEKF, several *local iterations* are made for each observation, wherein the observation equation is relinearized about the new state estimate. Thereby the IEKF provides faster convergence than the EKF, especially when the initial parameter estimate is far from the optimum.
3. Recast the IEKF as an iterated extended recursive least squares (IERLS) estimator. This is necessary for any practical implementation inasmuch as the state vector of the Kalman filters obtained in Steps 1 and 2 grows exponentially with time [2, §10.8].
4. The inverse of the state error covariance matrix  $P(t)$  is known as the *Fisher information matrix* [14, §3.4]. In the final step, the IERLS estimator is converted to *information form*; see Haykin [2, §10.9]. As will be discussed in Section IV-D, the information form of the estimator enables the diagonal loading of the resulting estimator to be periodically refreshed.

The steps above, as well as the final form of the IERLS estimator are summarized in [10, §6.3].

## C. Independent Processing of Subbands

One of the great advantages of conventional subband domain beamformers of either the LMS or RLS variety is that the sensor weights for the individual subbands can be set independently. This is a direct consequence of the statistical

<i>Input paired vector process:</i> $(\boldsymbol{\mu}(1), \mathbf{U}(1)), \dots, (\boldsymbol{\mu}(t), \mathbf{U}(t))$	
<i>Known parameters:</i>	
<ul style="list-style-type: none"> <li>• stacked blocking matrix: <math>\mathbf{B}</math></li> <li>• cepstral sequence obtained from GSC beamformer: <math>\mathbf{v}(t, \hat{\mathbf{W}}_a(t-1))</math></li> <li>• observation error covariance matrix: <math>\mathbf{Q}(t)</math></li> <li>• initial diagonal loading: <math>\sigma_0^2</math></li> </ul>	
<i>Initial conditions:</i>	
	$\hat{\mathbf{W}}_a(0) = \mathbf{0}; \quad \mathbf{P}^{-1}(0) = \sigma_0^2 \mathbf{I}$
<i>Computation:</i> $t = 1, 2, 3, \dots$	
	$\boldsymbol{\varphi}_1 = \hat{\mathbf{W}}_a(t-1)$
<i>Iterate:</i> $i = 1, 2, 3, \dots, f-1$	
	$\bar{\boldsymbol{\mu}}(t; \lambda^{-1/2} \boldsymbol{\varphi}_i) = \boldsymbol{\mu}(t) - \left[ \mathbf{v}(t; \lambda^{-1/2} \boldsymbol{\varphi}_i) - \sum_{m=0}^{M-1} \mathbf{c}_m(t; \lambda^{-1/2} \boldsymbol{\varphi}_{i,m}) \mathbf{Z}_m^H(t) \boldsymbol{\varphi}_{i,m} \right]$
<i>Iterate:</i> $m = 0, 1, \dots, M-1$	
	$\mathbf{P}_m^{-1}(t) = \lambda \mathbf{P}_m^{-1}(t-1) + \mathbf{Z}_m(t) \mathbf{c}_m^H(t; \lambda^{-1/2} \boldsymbol{\varphi}_{i,m}) \mathbf{Q}^{-1}(t) \mathbf{c}_m(t; \lambda^{-1/2} \boldsymbol{\varphi}_{i,m}) \mathbf{Z}_m^H(t) \quad (\text{IV.17})$
	$\mathbf{P}_m^{-1}(t) \boldsymbol{\varphi}_{i+1,m} = \lambda \mathbf{P}_m^{-1}(t-1) \hat{\mathbf{w}}_{a,m}(t-1) + \mathbf{Z}_m(t) \mathbf{c}_m^H(t; \lambda^{-1/2} \boldsymbol{\varphi}_{i,m}) \mathbf{Q}^{-1}(t) \bar{\boldsymbol{\mu}}(t; \lambda^{-1/2} \boldsymbol{\varphi}_i) \quad (\text{IV.18})$
	$\boldsymbol{\varphi}_{i+1,m} = \mathbf{P}_m(t) [\mathbf{P}_m^{-1}(t) \boldsymbol{\varphi}_{i+1,m}] \quad (\text{IV.19})$
<i>Set:</i>	
	$\hat{\mathbf{W}}_a(t) = \boldsymbol{\varphi}_f \quad (\text{IV.20})$
<i>Notes:</i>	
1. The local iteration over $i$ continues until there is no significant difference between $\boldsymbol{\varphi}_i$ and $\boldsymbol{\varphi}_{i+1}$ .	
2. $\mathbf{c}_m(t; \lambda^{-1/2} \boldsymbol{\varphi}_{i,m})$ is $m$ -th column of the matrix $\mathbf{C}_{\text{RLS}}(t)$ defined in (IV.3) and (IV.5) where $\mathbf{W}_a = \lambda^{-1/2} \boldsymbol{\varphi}_{i,m}$ .	

TABLE I

INFORMATION VERSION OF THE ITERATED EXTENDED RLS ESTIMATOR WITH UNCORRELATED SUBBAND STATE ESTIMATION ERRORS.

independence of the subbands [3, §5.2]. Treating each subband independently results in a tremendous savings in computation for the RLS variety beamformers, and much faster rate of convergence for LMS beamformers. In order to rewrite the recursions pertaining to the information version of the RLS estimator based on the assumption of uncorrelated subband state estimation errors, we assume  $\mathbf{P}^{-1}(t)$  is block diagonal, such that

$$\mathbf{P}^{-1}(t) = \text{diag} [\mathbf{P}_0^{-1}(t) \quad \mathbf{P}_1^{-1}(t) \quad \dots \quad \mathbf{P}_{M-1}^{-1}(t)] \quad (\text{IV.21})$$

and then define

$$\boldsymbol{\varphi}_i = [\boldsymbol{\varphi}_{i,0}^T \quad \boldsymbol{\varphi}_{i,1}^T \quad \boldsymbol{\varphi}_{i,2}^T \quad \dots \quad \boldsymbol{\varphi}_{i,M-1}^T]^T$$

The recursions defining the information version of the iterated extended RLS estimator can then be written as in Table I; see [10, §7.1].

#### D. Adding Diagonal Loading

Equations (IV.17–IV.18) can be solved for  $\mathbf{P}_m^{-1}(t)$  and  $\mathbf{P}_m^{-1}(t) \boldsymbol{\varphi}_{i+1,m}$ , whereupon the updated sensor weights follow from (IV.19). But the calculation of  $\mathbf{P}(t)$  requires an  $\mathcal{O}(N^3)$  inversion of  $\mathbf{P}^{-1}(t)$ . This expensive operation can be avoided by propagating the Cholesky decomposition [15, §4.2] or *square-root*  $\mathbf{P}_m^{-H/2}(t)$  of  $\mathbf{P}^{-1}(t)$  instead of  $\mathbf{P}^{-1}(t)$  itself, which implies the sensor weights  $\boldsymbol{\varphi}_{i+1,m}$  can be obtained through a simple forward substitution. This approach also

offers advantages with regard to numerical robustness. The details of the square-root implementation [2, §11] for the iterated extended RLS estimator considered here can be found in [10, §7.2].

In conventional beamforming, *diagonal loading* is usually added to  $\boldsymbol{\Phi}_m(t) = \mathbf{P}_m^{-1}(t)$  in order to restrict the size of  $\hat{\mathbf{w}}_{a,m}^H(t) = \boldsymbol{\varphi}_{f,m}^H$ . This is done to control the sidelobe structure and reduce the sensitivity of the beamformer to steering errors [3, §6.6]. Although  $\boldsymbol{\Phi}_m(t)$  is typically initialized as a diagonal matrix, this initial loading quickly decays whenever  $\lambda < 1$ . One advantage of the information version of the RLS estimator is that the diagonal loading can be periodically refreshed in the following fashion. Let  $\mathbf{e}_i$  denote the  $i$ -th unit vector. Suppose we would like to add the loading  $\beta_m^2(t)$  to the  $i$ -th diagonal component of  $\boldsymbol{\Phi}_m(t)$  according to

$$\boldsymbol{\Phi}_{L,m}(t) = \boldsymbol{\Phi}_m(t) + \beta_m^2(t) \mathbf{e}_i \mathbf{e}_i^T \quad (\text{IV.22})$$

This can be accomplished by forming the prearray

$$\mathbf{A} = [\mathbf{P}_m^{-H/2}(t) \quad \beta_m(t) \mathbf{e}_i]$$

and then constructing a unitary transformation  $\boldsymbol{\theta}_i$  that achieves [16]

$$\mathbf{A} \boldsymbol{\theta}_i = [\mathbf{P}_{L,m}^{-H/2}(t) \quad \mathbf{0}]$$

where  $\mathbf{P}_{L,m}^{-H/2}(t)$  is the desired Cholesky decomposition of (IV.22).

The application of each  $\theta_i$  requires  $\mathcal{O}(N^2)$  operations; hence, diagonally loading all diagonal components of  $\mathbf{P}_m^{-H/2}(t)$  is an  $\mathcal{O}(N^3)$  procedure. Note, however, that the diagonal loading need not be maintained at an exact level, but only within a broad range. Thus, with each iteration of RLS estimation, the diagonal components of  $\mathbf{P}_m^{-1/2}(t)$  can be successively loaded, so that the entire process remains  $\mathcal{O}(N^2)$ .

## V. SPEECH RECOGNITION EXPERIMENTS

The speech experiments described below were conducted with the Janus Recognition Toolkit (JRTk), which is developed and maintained jointly at the Universität Karlsruhe in Karlsruhe, Germany and at the Carnegie Mellon University in Pittsburgh, Pennsylvania, USA.

For these experiments, training was conducted on the English Spontaneous Scheduling Task (ESST) corpus, which contains approximately 35 hours of speech contributed by 242 speakers. The speech material was collected during spontaneous dialogs between two speakers engaged in the planning of overseas business trips. The dialogs were originally recorded with Sennheiser head-mounted, close-talking microphones. The clean speech from the original ESST recordings was used to train the HMM speech recognizer for all experiments.

The ESST test set is approximately 3.5 hours in length and contains 58 dialog halves contributed by 16 unique speakers. The total test set comprises 22,889 words. For the beamforming experiments described below, the clean speech test data was played through a loudspeaker into a  $6 \times 8$  meter room with a reverberation time of approximately 350 ms, and then recorded with a linear microphone array. The array consisted of eight Sennheiser omnidirectional microphones separated by 4.1 cm. The stationary loudspeaker was located 2 m in front of one end of the array. In addition to the source speech data, interference data was also recorded by moving the speaker parallel to the array approximately 3 m from the location used for the source. Two types of interference were recorded: music from a chamber orchestra and speech from another talker.

All speech data was digitally sampled at a rate of 16 kHz. The speech features used for all experiments were obtained by estimating 13 cepstral components, along with their first and second differences. Features were calculated every 10 ms using a 20 ms sliding window. In extracting the cepstral features, a Hamming window 320 samples in length was first applied to each segment of speech. The windowed segment was then padded with zeros and a 512 point FFT was calculated. For the single-channel experiments, the frequency samples were then used to calculate cepstral coefficients as in (III.2–III.3). For the various beamforming scenarios, the frequency samples from each sensor in the array were combined as in (III.6) and thereafter the cepstral coefficients were calculated.

All experiments with HMM beamforming algorithms were conducted with *oracle* state alignments, which were obtained by performing a Viterbi state alignment with the correct transcription on the output of the delay and sum (D&S) beamformer. All sensor weight optimization was performed in the cepstral domain as described in Sections III and IV.

Beamformer	Interference		
	None	Music	Talking
Single-Channel	61.37	65.94	64.36
D&S	51.34	59.95	59.39
GSC	54.72	58.24	56.17
Unconstrained	51.08	58.51	57.88

TABLE II

WORD ERROR RATES FOR GLOBAL (I.E., SINGLE UTTERANCE) ML BEAMFORMING EXPERIMENTS.

$\alpha^2$	Interference		
	None	Music	Talking
D&S	51.34	59.95	59.39
$10^{-4}$	51.31	59.84	59.39
$10^{-3}$	51.32	59.70	59.25
$10^{-2}$	51.17	59.54	58.94
$10^{-1}$	51.27	59.41	58.76
$10^0$	51.28	59.42	58.68

TABLE III

WORD ERROR RATES FOR HMM-LMS BEAMFORMING EXPERIMENTS.

An unadapted HMM with 48 Gaussians for each of 2,340 codebooks was used for all speech recognition experiments. Using this model to recognize the original clean speech test set, a word error rate (WER) of 31.94% was achieved. For the HMM beamforming experiments, an auxiliary model was used for sensor weight optimization, which had a single Gaussian component per codebook. Only static cepstral features were used for sensor weight estimation.

### A. Experimental Results

The results of the *global* optimization experiments, wherein the longest utterance for each speaker was selected, and multiple iterations of conjugate gradient descent were run on this one utterance to find the ML sensor weights, are shown in Table II. As is clear from the table, the global sensor weight optimization provided only a marginal gains in the experiments without interference. For the cases with music and talking interference, however, the reduction in WER with respect to the D&S baseline was substantial.

The results of the experiments with the HMM-LMS beamformer are given in Table III. The left column indicates the value assigned the bound  $\alpha^2$  on the size of the active weight vector; see (III.18) and (III.20). As is clear from the table, no improvement in recognition performance was observed for the no interference case. For the music and talking interference cases, there is a statistically significant reduction in WER, but the gains are *not* so large as those obtained with the global optimization scheme.

The results of the experiments with the HMM-RLS beamformer are given in Table IV. The left column indicates the amount of diagonal loading applied as in (IV.22) measured with respect to the average power in all subbands; more loading implies the active weight vector is smaller. To achieve fast initial convergence, five (5) local iterations with a forgetting factor of  $\lambda = 0.98$  were conducted for each of the first 1000 cepstral observation according to the iterated extended RLS

Load Level	Interference		
	None	Music	Talking
D&S	51.34	59.95	59.39
-35	51.32	59.79	59.20
-40	51.25	59.35	58.94
-45	52.27	60.38	58.77
-50	58.35	71.24	65.54

TABLE IV

WORD ERROR RATES FOR ITERATED EXTENDED RLS BEAMFORMING EXPERIMENTS.

Load Level (dB)	Interference	
	None	Music
D&S	51.34	59.95
50	51.31	59.97
45	51.28	60.07
40	51.31	60.21
35	51.46	60.77
30	53.68	63.36

TABLE V

WORD ERROR RATES FOR THE CONVENTIONAL RLS BEAMFORMER IN GSC CONFIGURATION.

algorithm described in Table I. As the sensor weights were largely stable at that point, one a single local iteration with  $\lambda = 1.0$  was performed thereafter. Here again we observe no gain with the no interference case. For the music and talking interference cases, there is a statistically significant reduction in WER for diagonal load levels of -40 and -45 dB respectively, but the gains are *not* so large as those obtained with the global optimization scheme.

Finally, for the sake of comparison, we tested conventional RLS and LMS beamformers with the same Hamming window-FFT filterbank used for the HMM beamforming experiments. The results of the experiments are given in Tables V and VI respectively. Frequency dependent scaling factors as in (III.19) were used in estimating the sensor weights for the conventional LMS beamformer. For both types of conventional beamformers, the variations in WER are essentially random; i.e., the conventional algorithms failed to provide a statistically significant improvement in system performance.

### B. Discussion

As noted before, the global optimization procedure provided only a marginal reduction in WER with respect to D&S in the

$\alpha^2$	Interference	
	None	Music
D&S	51.34	59.95
$10^{-4}$	51.32	59.92
$10^{-3}$	51.47	59.82
$10^{-2}$	51.72	60.37
$10^{-1}$	54.96	64.90

TABLE VI

WORD ERROR RATES FOR THE CONVENTIONAL LMS BEAMFORMER IN GSC CONFIGURATION.

no interference scenario, but substantial reductions for both music and talking interference. We attribute this difference in behavior primarily to the short analysis window of 20 ms used for these experiments. Such a short window is adequate for an adaptive beamformer to suppress a *direct* interference signal, but is insufficient to remove reverberation, which is the primary distortion in the absence of any strong interferer. It is worth noting that Seltzer [1] obtained significant reductions in WER for speech material that had been artificially “reverberated” by convolution with a room impulse response. This was achieved, however, by including *delayed* frequency samples in the beamforming procedure. As discussed in the next section, we believe that the same effect can be obtained by using a better filter bank, with a longer memory.

Based on the global optimization experiments, it seems that cepstral-domain optimization is very viable. Seltzer [1] achieved substantial reductions in WER by performing all parameter optimization in the log-spectral, instead of cepstral domain. We also conducted several experiments wherein sensor weight optimization was performed in the log-spectral domain. Unfortunately, none of these experiments resulted in a WER reduction with respect to the D&S beamformer. Our efforts to resolve this discrepancy are ongoing.

The results in Table II clearly show that the constrained GSC actually outperforms the fully unconstrained beamformer. One possible explanation for this observation is the following: Whenever a random variable or vector transformed, it’s likelihood in the transformed space must be multiplied by the *Jacobian* of the transformation [17, §6.3] to obtain the true likelihood. In the present case, a transformation is performed on the spectral input of each sensor to obtain the final output of the beamformer. Because the GSC imposes a distortionless constraint on the look direction, however, the Jacobian for the look direction is unity. The unconstrained beamformer imposes no such constraint, but neglects to account for the contribution of the Jacobian during sensor weight optimization. Hence, the “likelihood” used by the unconstrained beamformer is not the actual likelihood of the source features.

As reported in Table IV, we have observed statistically significant WER reductions with the adaptive RLS-HMM beamformer, but not so much as with the global optimization schemes. We hypothesize that the smaller reduction in WER is a product of the failure of the Hamming window-FFT filter bank to provide sufficiently independent spectral samples, as was explicitly assumed in (IV.21); i.e., there is simply too much overlap between adjacent bins in the frequency domain due to the spectral smearing of the relatively short analysis window. The fact that the conventional algorithms provided no improvement with respect to the D&S beamformer would seem to support this hypothesis. Further experiments with a better filter bank, however, are required to verify it. As the design of perfect reconstruction filter banks is now a well-established [18] field, there are several well-proven designs that might potentially be used in the current application.

## VI. CONCLUSIONS AND FUTURE WORK

In this work, we have briefly discussed the meeting room scenario as the next challenge for the ASR community. Also,

following [1] we have considered the problem of adaptively setting the sensor weights of a GSC beamformer under a ML criterion. In particular, we showed that it was straightforward to develop both LMS and RLS adaptive algorithms for ML beamforming. These algorithms had many of the same advantages and drawbacks of conventional LMS and RLS algorithms: Both LMS and RLS algorithms update the sensor weights each time a new subband snapshot is received, and both process each snapshot only once, and thereafter discard it. The LMS algorithm developed here, much like the conventional LMS algorithm, is characterized by simplicity, both conceptually and in its implementation. Moreover, it has a computational complexity of  $\mathcal{O}(N)$  where  $N$  is the number of sensors in the array. Unfortunately, this simplicity is offset by the potentially slow convergence of the LMS algorithm [3, §7.7]. The RLS algorithm is more difficult conceptually and has a computational complexity of  $\mathcal{O}(N^2)$ . But in return for this higher computational load, the RLS algorithm provides faster convergence [3, §7.4]. In the speech recognition experiments described in Section V, both LMS and RLS versions of the HMM beamformer provided statistically significant reductions in word error rate with respect to D&S. These reductions, however, were not so large as those obtained with global optimization algorithm, wherein multiple iterations of conjugate gradient descent are performed on a single utterance until the sensor weights converge.

Although this work has provided a relatively detailed discussion of the parameter estimation aspects of the ML beamforming problem, it has neglected one very important issue: A beamformer operating in a reverberant environment must operate on blocks of speech on the order of 300-400 ms in length. As mentioned in the introduction, this follows from the well-known fact that the primary hindrance to the accurate recognition of far field speech is reverberation, which tends to smear the temporal characteristics of the speech signal, and thereby cause devastating degradations in ASR performance. In order to effectively compensate for reverberation, each channel of a filter-and-sum beamformer must have a filter length comparable to, or preferably longer than, the reverberation time of the room in which the ASR system operates. For a medium-sized meeting room, the reverberation time can easily be a few hundred milliseconds. An ASR system, on the other hand, must operate on speech segments on the order of 15-20 ms because, for such relatively short segments, the articulators and hence the characteristics of the resulting speech are *quasi-stationary*, which is not true of longer segments. The problem outlined above is exactly that which we hope to consider in future: How can the conflicting signal processing requirements imposed by the beamforming and recognition components of a medium field ASR system be effectively resolved? Indeed, our initial efforts to answer this question can be found in [19].

A well-known limitation of conventional RLS estimators is the assumption that the parameters to be estimated change either not at all, or only very slowly. Indeed, Haykin [2, §14.6] explains that because of this assumption, a conventional RLS estimator actually has *worse* tracking capability than the simpler LMS estimator, despite its better convergence performance. As a beamformer operating in a real meeting

room environment must be able to adapt to moving sources and other changes in the environment, this limitation in tracking is clearly a drawback. Fortunately, one possible remedy is readily obtained by exploiting the equivalence of RLS estimators and Kalman filters mentioned in Section IV-B. Because the RLS estimator implicitly assumes the parameters to be estimated are deterministic and constant, casting the linearized RLS estimator developed in Section IV-A as a Kalman filter provides a filter with no *process noise*. For applications wherein these parameters change with time, however, it is straightforward to include process noise in the model. The larger the covariance of this process noise, the greater the uncertainty in the estimate of the state parameters. Hence, the covariance of the process noise can be allowed to vary with time, such that during intervals wherein the speaker's position or other environmental factors change quickly, the state parameters are also allowed to quickly adapt. Haykin [2, §14.7] discusses several applications where this approach was used with good effect.

## REFERENCES

- [1] M. Seltzer, "Microphone array processing for robust speech recognition," Ph.D. dissertation, Carnegie Mellon University, 2003.
- [2] S. Haykin, *Adaptive Filter Theory*, 4th ed. New York: Prentice Hall, 2002.
- [3] H. L. Van Trees, *Optimum Array Processing*. New York: Wiley-Interscience, 2002.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39 B, pp. 1–38, 1977.
- [5] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge: MIT Press, 1997.
- [6] D. M. Titterton, "Recursive parameter estimation using incomplete data," *J. Roy. Stat. Soc. B*, vol. 46, pp. 256–267, 1984.
- [7] L. Frenkel and M. Feder, "Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking," *IEEE Trans. Signal Processing*, vol. 47, pp. 306–320, Feb. 1999.
- [8] J. Deller, J. Hansen, and J. Proakis, *Discrete-Time Processing of Speech Signals*. New York: Macmillan Publishing, 1993.
- [9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge: Cambridge University Press, 1992.
- [10] J. McDonough, D. Raub, M. Wölfel, and A. Waibel, "Towards adaptive hidden Markov model beamformers," Interactive Systems Labs, Universität Karlsruhe, Tech. Rep. 101, October 2003.
- [11] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, pp. 18–60, July 1994.
- [12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, pp. 35–45, 1960.
- [13] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.
- [14] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. New York: John Wiley & Sons, 1984.
- [15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: The Johns Hopkins University Press, 1996.
- [16] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations," *Mathematics of Computation*, vol. 28, no. 126, pp. 505–535, 1974.
- [17] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill Publishing, 1984.
- [18] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs: Prentice Hall, 1993.
- [19] J. McDonough, "Filter bank design for hidden Markov model beamformers," Interactive Systems Labs, Universität Karlsruhe, Tech. Rep. 103, December 2003.

**John McDonough** received his Bachelor of Science in 1989 and Master of Science in 1992 from Rensselaer Polytechnic Institute. From January 1993 until August 1997 he worked at the Bolt, Beranek, and Newman Corporation in Cambridge, Massachusetts primarily on large vocabulary speech recognition systems. In September 1997 he began doctoral studies at the Johns Hopkins University in Baltimore, Maryland, which he completed in April 2000. Since January of 2000 he has been employed at the Interactive Systems Labs at the Universität of Karlsruhe in Karlsruhe, Germany as a researcher and lecturer.

**Dominik Raub** is currently a student of computer science at the Universität Karlsruhe in Karlsruhe, Germany. He spent the 2002/2003 academic year on a Fulbright scholarship at the Carnegie Mellon University in Pittsburgh, PA, USA, working on signal processing, particularly beamforming for application in speech recognition. He hopes to complete his Diploma degree by Summer 2004.

**Matthias Wölfel** has received his Diplom in Electrical Engineering and Information Technology in 2003 from the Universität Karlsruhe in Karlsruhe, Germany. From September 2000 until June 2001 he participated in an exchange program with the University of Massachusetts, Dartmouth. In September 2002 he went to the Carnegie Mellon University in Pittsburgh, Pennsylvania as a visiting researcher for a period of four months. Since March 2003 he has been pursuing doctoral studies in the field of Computer Science at the Interactive Systems Laboratories at the Universität of Karlsruhe.

**Alex Waibel** is a Professor of Computer Science at Carnegie Mellon University, Pittsburgh and at the Universität of Karlsruhe (Germany). He directs the Interactive Systems Laboratories ([www.is.cs.cmu.edu](http://www.is.cs.cmu.edu)) at both Universities with research emphasis in speech recognition, handwriting recognition, language processing, speech translation, machine learning and multimodal and multimedia interfaces. At Carnegie Mellon, he also serves as Associate Director of the Language Technology Institute and as Director of the Language Technology PhD program. He was one of the founding member of the CMU's Human Computer Interaction Institute (HCII) and continues on its core faculty.

Dr. Waibel was one of the founders of C-STAR, the international consortium for speech translation research and served as its chairman from 1998-2000. His team has developed the JANUS speech translation system, the JANUS speech recognition toolkit, and a number of multimodal systems including the meeting room, the Meeting recognizer and meeting browser.

Dr. Waibel received the B.S. in Electrical Engineering from the Massachusetts Institute of Technology in 1979, and his M.S. and Ph.D. degrees in Computer Science from Carnegie Mellon University in 1980 and 1986. His work on the Time Delay Neural Networks was awarded the IEEE best paper award in 1990; his work on multilingual and speech translation systems the "Alcatel SEL Research Prize for Technical Communication" in 1994, the "Allen Newell Award for Research Excellence" from CMU in 2002 and the Speech Communication Best Paper Award in 2002.